

Student Name: Thanh Tam Vo
Student ID: 103487596

COS20007 Object Oriented Programming -Task 7.1P

Object Oriented Programming is a model to organize artefact (or object) by its characteristic, data, and behaviors. With OOP, making the interaction between those objects is more effective since objects are divided into small group for developer to manage easily. Talking about OOP, the first thing, which is mentioned, is “*object*” or in OOP term, we prefer the word “*class*”, each class has its own attribute and functionality.

Eg: A class named Bird has an attribute called ‘name’ and a method called ‘fly’. In daily life, a bird has its name, and it can fly, so as we can see, the object “bird” can be observed under the OOP angle.

More importantly, there are four signature things in OOP: Encapsulation, Inheritance, Polymorphism, Abstraction.

- **Encapsulation:** is the process of gathering all the data and methods, which are implemented using those data, packing all of them into a *class*, the class *Bird* which is mentioned above is an example of encapsulation.
- **Inheritance:** Like human being, a class has its ancient and it can inherit from the parent class and can behave like the same way as its parent do. We call a child class as a derived class.

Eg: A class dog inherit from a class called mammal, so the class dog has a warm-blooded characteristic (because the mammal is warm-blooded)

- **Abstraction:** Abstraction is the term to present the necessary information of a class, but not actually show how the information is implemented
Eg: a class “washing machine” have a function Wash, but that function will not be showed how it work

```
using System;
namespace WashingMachine
{
    public abstract class WashingMachine
    {
        public WashingMachine()
        {
        }

        public abstract void Wash();
    }
}
```

Figure 1. An abstract class called washing machine

- **Polymorphism:** Many derived classes from one given abstract class, can have their own way to perform a method. From a single function from the abstract class, it can be performed differently depending on the child class.
Eg: a class called SamsungWash and a class called AppleWash, which inherit from the same class WashingMachine, can perform the Wash method differently.

OOP Terminology

There are some important keywords in OOP concept:

- **private:** a private attribute of a class can only being accessed by that class, to make an interaction with that attribute, we need a property from that class.
- **public:** beside from private attribute, we have public attribute allows other class can interact with the public attribute with the current class.
- **override:** an override method is method that is already modified by the developers. Usually, we see override method in a class which inherit from an abstract class.
- **virtual:** a virtual method can be seen from a base class (not only in abstract class), however, it can be modified in the derived class
- **interface:** it is the same as an abstract class but without body. It can not create an object, but a class can inherit from multiple interface (Note: a class can only inherit from one class, this will make the interface and an abstract class different)

RoadMap

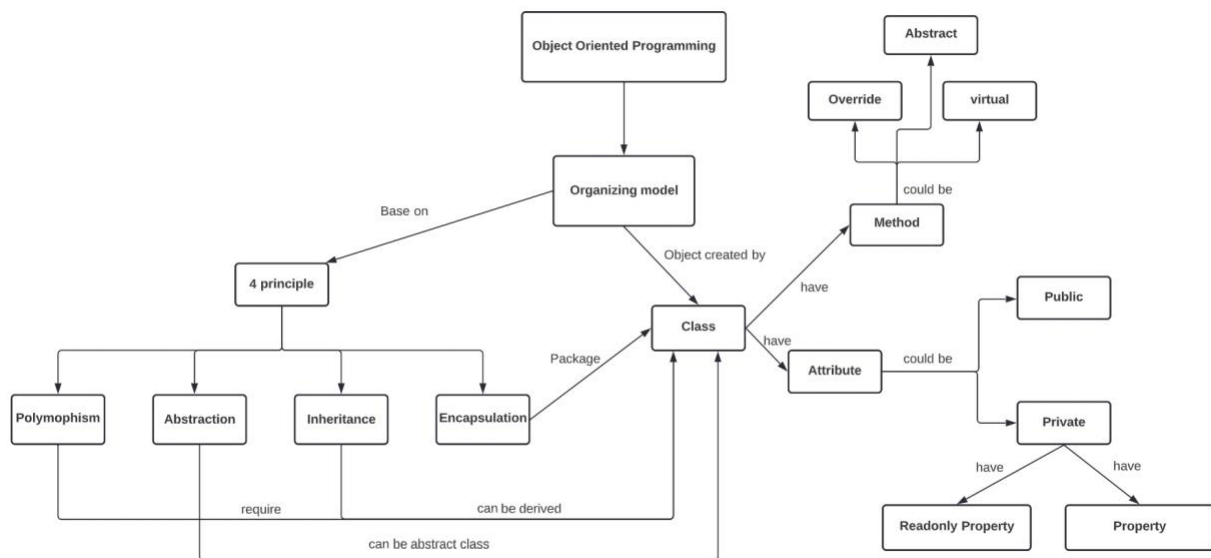


Figure 2. Roadmap for OOP base on the above explained concept